View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Repositorio Institucional de la Universidad de Alicante

International Journal of Artificial Intelligence & Applications (IJAIA), Vol.2, No.4, October 2011

ROBOTIC CONTROL BASED ON THE HUMAN NERVOUS SYSTEM

Jose Vicente Berna-Martinez[1] and Francisco Macia-Perez[2]

[1]Department of Computer Science, University of Alicante, Alicante, Spain
jvberna@dtic.ua.es
[2]Department of Computer Science, University of Alicante, Alicante, Spain
pmacia@dtic.ua.es

ABSTRACT

This article presents a model of robotic control system inspired by the human neuroregulatory system. This model allows the application of functional and organizational principles of biological systems to robotic systems. It also proposes appropriate technologies to implement this proposal, in particular the services. To illustrate the proposal, we implemented a control system for mobile robots in dynamic open environments, demonstrating the viability of both the model and the technologies chosen for implementation.

KEYWORDS

Bioinspired system, robotic system control, SOA, autonomous robots, distributed autonomous systems.

1. INTRODUCTION

Robotics is a field in which converge a number of factors such as the rapid evolution of hardware and software, the need for interdisciplinary developer, the great diversity of tasks and missions to be performed or the social, economic and political factors that surround the robotic [1]. These factors are multiplied by the evolution in the use of robots. Traditionally robots have been used in industrial environments perfectly controlled, static, and designed especially for the development of robotic activity. At present, robots are being widely used for a multitude of other professional or personal tasks in dynamic and open environments, and they are used by non expert users [2]. This implies that the number and variety of robotic systems is growing constantly and the absence of models, methodologies, technologies, standards or widely available homogeneous tools implies that these systems are often solved in ad-hoc mode, becoming more distant from each other with incompatibilities, lack of interoperability and difficult to maintain medium and long term, because hardware and software are tightly coupled and any modifications or alterations of systems are expensive and require reprogramming all or part of the system [2].

It is therefore necessary to propose innovative approaches to keep decoupled the physical part of the logic in a robot, enabling the rapid and flexible creation of complex robotic systems while maintaining scalability and flexibility for adaptation and maintenance, posibility of interaction of robot with real environments and other robotic systems, and of course, allow robots exhibit intelligent behavior, in other words, a robot that behaves like a human performing the same task [2], all this maintaining a structure and philosophy common to all the elements that form a robotic system regardless of their complexity [3]. With this objective in this paper we propose a model of control systems inspired by the functioning and organization of human neuroregulatory system and its implementation through the use of services. The combination of both approaches allows us to merge the advantages of complex control software along with the benefits of the paradigm of services.

DOI : 10.5121/ijaia.2011.2409

Then, this paper presents the following sections: Section 2 is the state of the art where we will review several major proposals around bio-inspired control systems and the main philosophy of operation and modeling of biological systems, in the section 3 we develop the proposed control system model inspired by the functioning and organization of human neuroregulatory system, in Section 4 we justify the paradigm selected for software implementation, in Section 5 we perform the instantiation of the model for navigation control system in autonomous mobile robot, in section 6 shows the analysis of test results and finally, at Section 7 we present the main conclusions and future lines of work for this article.

2. BACKGROUND

A robotic system consists of a physical system and control system. The control system must act on the physical part to achieve an end, such as move on land, making a precise welding, collect objects or cleaning in a given area. There is a huge set of approaches to design a control system which should capture the state of the world, decide what action to take and act on the physical resources available. One approach currently being used is inspired by biology because the biological neuroregulatory system is a control system of a physical body in similar circumstances showing by robotic system [4], and, above all, because the control system objective is to show a robot with intelligent behavior, understanding intelligence as a human-like behavior [5]. There are two great visions when we characterize the biological system: deliberative or reactive. Reactive vision makes a decision based on the interpretation of the real world at every moment. The deliberative approach faces a problem reasoning about a model of the world, rules and representations of knowledge to prepare for a decision.

In reactive systems, the main problem is action selection mechanism selection (ASM), ethology traditionally defines this problem as the problem of switching behavior or action selection [6]. En un robot, esto significa que varios módulos de comportamiento se desarrollan y el comportamiento adecuado se selecciona utilizando las circunstancias o los estados del mundo[7]. Such mechanisms allow emerging opportunistic behavior that is not encoded in the control logic but arises from the interaction of the modules of behavior and selection [8]. This approach allows taking into consideration such profound behavioral aspects such as ethics [9] and practical engineering aspects such as using modern techniques for implementation in the form of distributed system [10], thus enabling the system scalability and distribution.

Through a deliberative approach we can represent the knowledge using structures, memories, rules or models. This vision has allowed to build frameworks for knowledge representation such as ACT-R [11], Soar [12], Icarus [13] o PRODIGY [14]. One of the central issues in this type of system is how to access each type of information as the state of the environment, decisions, memory or learning, which characterizes and differentiates each proposal. Another major difference in each system is how to implement the skills, that is, if you use embedded processors that perform a function or if these features are contained in the form of knowledge that is part of the system elements.

Deliberative architectures have advantages in predictable environments, but require an accurate model of the world, actions and their effects so that systems can deliberate and make decisions [14]. Reactive architectures are advantageous in dynamic and unpredictable environments, but require great knowledge to make a decision in any situation and not all are known for their design [8]. To join the two positions, hybrid architectures have been developed.

Hybrid architectures use various separate layers to develop reactive or deliberative functions. For example, [15] presented a hybrid architecture that defines 4 levels: decision, behavior, analysis and communications hardware. [16] provides an architecture designed in three layers: deliberative, reactive and level of communications. [17] uses a split architecture into two layers, deliberative layer responsible for planning and reactive layer responsible for implementing the planned actions. Each proposal requires a division into levels based on the problem they are solving.

All these proposals make clear the appropriateness of seeking mechanisms similar to biological control to develop robotic systems that seek an approach to human behavior, but these proposals produce incompatibilities and far systems between. Also show the fundamental characteristics of biological systems, the capacity for deliberation and reaction, leading to the conclusion that both mechanisms must be integrated to achieve more complex behaviors that require both trends. The study of the different proposals leads to the following general characteristics of robotic control systems:

- Control applications must be modular to allow code reuse and rapid development.
- The control logic must be independent of hardware. The hardware provides the possibilities, but the software develops the skills.
- Support for communications should be provided by framework in which is developed the system and details should be hided for implementation of intelligence of the robot.
- Components must be able to communicate asynchronously transmitting values. If the components use references cannot be distributed independently.
- The components must be able to be linked dynamically, using modules that are necessary even in runtime.
- Reactive techniques exploit the characteristics of the real environment
- Deliberative techniques allow us to infer knowledge that is not implicit in the environment

These bio-inspired systems are based on the study of biological behavior and structure of biological nervous system. Focusing on the human neuroregulatory system, the nervous system can be described as a complex network of neural structures that control the activity of the organism. Nervous system functions are to collect, process and transmit nerve signals through different structures in order to control both somatic and autonomous activities, and also, some activities are completely reactive and we have no control over them, while for others there may be some or all reasoning. At first glance, the activities of the nervous system may seem contradictory. For example, the activity of the sympathetic and parasympathetic system. These systems are responsible for the activation or relaxation of internal activity, but we require the action of both systems for correct visceral activity and the internal environment [18].

The nervous system is composed of different nerve centers scattered throughout the entire body. Every center produces states of regulation more or less complex. Each of these centers also has an independent activity, and each state of regulation are hierarchical one over the other. Less evolved centers are located in the periphery while the more evolved centers are located at the central level [19]. The nervous system is formed by a process of evolution that has lasted thousands of years. During this process control centers have been added to neuroregulatory system to modulate, monitor, enhance, inhibit, suppress or substitute the underlying functionality [20][21]. This development occurs incrementally, adding elements to the nervous system and specializing nerve areas [22]. These new control centers have also been organized as new layers of the nervous system [23].

In addition to the nerve centers, the nervous system are also affected by the hormonal brain. In real life, the diffuse influence of this system, called neuromodulation, modifies the intensity and accuracy of the actions of the control centers and changes the final behavior of system [24].

Therefore, the sum of all the influences of regulatory structures is what triggers a behavior, action, reaction or stabilization of the entire system without having a specific center

commissioned to produce an action. The interaction of all these and the sum of its influences is what causes global behavior [18]. The nervous system has a distributed and complex nature, where each element carries out its autonomous control, producing an emergent behavior resulting from the sum of the actions of each system element. In addition, the activity of the control centers can be changed due to the influence of neurotransmitters.

To model this behavior is necessary to use paradigms that bring us wealth expressive enough to reflect all the characteristics described. In this context, the agent paradigm offers a high level of abstraction appropriate to address the complexity of the problem [16][25][26][27]. Multi-agent systems provide a framework capable of providing sufficient expressive power to address the modeling of these distributed systems, taking into account the emergent behavior and the possibility of modifying the structure of the model because we get new developments or knowledge of the system, either by technological innovations or advances in research. Although there is no agreement about the exact definition of agent, exists consensus on some of the properties that must be met [28][29][30n]: autonomy, must be independent, proactive and a set of objectives; learning, ability to adapt to the environment; reasoning, may be reactive, proactive or hybrid; memory, ability to recall and reasoning necessary for learning; socializing, be able to cooperate not only with other agents; communication, ability to get along with other agents, to perceive the environment and act on it; safety, ensure proper operation, avoiding inappropriate interactions and respond appropriately to events unknown. Looking at these characteristics, these are common to those exhibited by the control centers of the nervous system. Besides these properties, the agents may have other attributes such as mobility, modularity, scalability and reliability, although not strictly necessary. Our model uses these properties to describe the operation of the robot control system like a human nervous system.

3. MODELLING ROBOTIC CONTROL SYSTEM BASED ON HUMAN NEUROREGULATORY SYSTEM

Our proposal is that a robot can be controlled by a set of independent control centers that operate and are organized similarly to the human nervous system, causing many influences on the system and the sum of all these causes a complex emergent behavior. So, we model the robot as a neuroregulatory control system.

The human neuroregulatory system consists of a series of autonomous regulating centers located throughout the nervous system. These control centers receive information from the biological sensing elements (afferent signals, *AS*) and process that information to produce new signals (internal signals, *IS*) that are retransmitted to other centers. After several stages of processing, response signals are produced (efferent signals, *ES*) and sent to the elements of action [31] [32].

Our model will reflect this operation through a system consisting of agents, which we call functional elements or functional entities (*fe*) and that each of these entities provide functionality to the system. Each center is modeled as an independent agent that performs a function as a biological control center.

Based on the action and reaction system described in [25] and the model of neuroregulatory lower urinary tract [32] we can describe the elements that form a robotic system using the structure:

$$RS=\langle MS, RRS, {}^{MS}I_{RRS}\rangle \qquad (1)$$

RS represents the complete Robotic System, *MS* defines the Mechanical System, *RRS* the Regulatory Robotic System consists of all functional entities ${}^{MS}I_{RRS}$ represents the interface between both systems, basically the complex system of signal connections and afferent and efferent internal.

The interface is defined by the structure:

$${}^{MS}I_{RRS}=\langle \Sigma, \Gamma, P \rangle \tag{2}$$

Where Σ represents the set of possible states of the system. Γ identifies the set consists of the possible intentions of actions in the system by the functional entities. As mentioned already, the functional entities do not have complete control of the system and have to combine their objectives. The result of each action is represented as an intention of acting on the system. And finally, P is the set of all possible actions of the regulatory system that can be performed on the mechanical system to change the status of the robot.

Each functional entity tries to change the system state by taking actions on it. These attempts to influence or action is defined as:

$$\Gamma=\langle \gamma_1, \gamma_2, ..., \gamma_n \rangle \tag{3}$$

Where every γ_i is a list of pairs consisting of a signal and its value, ie:

$$\gamma_i=\langle (sig_1,val_1), (sig_2,val_2), ..., (sig_{Card(C)},val_{Card(C)}) \rangle \tag{4}$$

In this case, the set C corresponds to the union of the sets of internal and efferent signals that circulate in the system $IS \cup AS$. When a Control Center does not want to change the system, bring the empty influence γ_0. This influence will act as the neutral element of the set Γ and can be carried by any *fe* when does not want to change the system state

Functional entities must take action and these actions change the state of the system. Entities accomplish this executing actions. The set of all possible actions that can be performed on a system can be defined as "(5)" and each of the actions described as "(6)".

$$P=\langle p_1, p_2, ..., p_k \rangle \tag{5}$$
$$P=\langle name, pre, post \rangle \tag{6}$$

name is an expression equal to $f\,(x_1, x_2, ..., x_k)$ and each x_i is a variable authorized to appear in *pre* and *post* formulas. *pre/post* formulas are sets of type $g\,(a_1, a_2, ..., a_n)$ where g is a predicate of n variables and each a_i are constants or variables. *pre* describe the conditions that must be verified to perform the action and *post* refers to the set of influences that occur when executed actions. We define two actions in our control system: p_s and p_0.

$$p_s=\langle SetSignalValue(),True(),Value() \rangle \tag{7}$$
$$p_0=\langle EmptyTask(),True, \gamma_0 \rangle \tag{8}$$

SetSignalValue() take as input a list τ_i and as output the results of the formula *Value()*.The formula *Value()* sets the new value of each signal pair indicated in the list τ_i specified in the action. The action p_0 defines the empty action and will act as a neutral regarding the status of the system. This action can always be out and will not alter the state of the system and provides the empty influence.

Because all the functional elements of the system act simultaneously as in the human neuroregulatory system, different influences occur simultaneously and therefore we define the union of these influences $\cup\!\!\int$ as the function that combines the influences and contributions of every *fe*. This feature provides an array of influences from the influences that each *fe*: $\cup\!\!\int : \Gamma^n \rightarrow \Gamma$.

The set of all functional entities of the robotic system form the RRS. All entities are responsible for controlling physical and cognitive activity, like the nervous centers control a biological

system. Each *fe* receives a number of afferent signals (AS_{fe}), these afferent signals may come from the *MS* or other *fe*, and processes the AS_{ef} and relays the results (efferent signals ES_{fe}) to other functional elements or mechanical system. The set of all functional entities that make up the robotic controller is defined as:

$$RRS=\langle fe_1, fe_2, ..., fe_n \rangle \quad (9)$$

Every *fe* is represented by a PDE architecture (perception-deliberation-execution) to which is incorporated memory capacity to be able to maintain its internal state and obtain a similar performance to the biological. The structure of each functional entity will be described using the structure:

$$fe=\langle \Phi_{fe}, S_{fe}, Percept_{fe}, Mem_{fe}, Decision_{fe}, Exec_{fe} \rangle \quad (10)$$

In this structure Φ_{fe} is the set of perceptions, S_{ef} is a set of internal states, $Percept_{fe}$ provides system status information, Mem_{fe} gives the ability to be aware of their own state, $Decision_{fe}$ selects the next task to execute; $Exec_{fe}$ represents the intention that has the functional entity to act on the system.

For a *fe*, the perception is the ability to sort and distinguish system states that it regulates, the information that interests him. Perception is defined as a function that associates a set of values, called perceptions or stimuli, with a set of system states $Percept_{fe}: \Sigma \rightarrow \Phi_{ef}$, so the perception is associated with the possible states of the system and is expressed as $\Phi = Percept(\sigma)$.

The set of possible perceptions associated with a particular *fe* is defined as $\Phi_{fe}=\langle \phi_1, \phi_2, ..., \phi_n \rangle$, and each ϕ_i is composed of a list of pairs (signal, valueSignal) as defined above, and we can define the empty perception ϕ_0 as a list of null pairs that will occur when *fe* is no destination of any *AS* or origin of any *EF*.

Each functional entity has an internal state that can remember, which allows more complex behaviors. The set of internal states of a functional entity is defined as $S_{fe}=\langle s_1, s_2, \ldots, s_n \rangle$. In the case of our robot controller system consists of a list of pairs (signal, valueSignal) of all internal signals of *fe*.

The decision function defines a perception task using the system status and experience in the past, determined by its internal state $Decision_{ef}: \Phi_{ef} \times S_{ef} \rightarrow P$, so the action selected by the functional entity is defined as $p=Decision(\phi,s)$. Given the actions previously defined for the system, $Decision_{fe}$ can return p_0 or p_s:

$$Decision_{fe}(\phi,s)=(SetSignalValue(FunD_{fe}(\phi,s)) \quad \text{if } PreD_{fe}(\phi,s)=True$$
$$Decision_{fe}(\phi,s)=p0 \quad \text{if } PreD_{fe}(\phi,s)=false \qquad (11)$$

$PreD_{fe}(\phi,s)$ defines the precondition that must be satisfied to run SetSignalValue(), using perception and internal state $PreD_{fe}:\Phi_{fe}\times S_{fe}\rightarrow Boolean$, and $FunD_{fe}(\phi,s)$ associates a perception and internal state with the influence of the functional entity in the system, $FunD_{fe}: :\Phi_{fe}\times S_{fe}\rightarrow\Gamma$.

The memory function associates an internal state of the functional entity with its current perception of the environment and past experience $Mem_{fe}:\Phi_{fe}\times S_{fe}\rightarrow S_{fe}$. The memory function, like the decision, act when it meets a precondition:

$$Mem_{fe}(\phi,s)=FunM_{fe}(\phi,s) \quad \text{if } PreM_{fe}(\phi,s)=True$$
$$Mem_{fe}(\phi,s)=s_0 \quad \text{if } PreM_{fe}(\phi,s)=False \qquad (12)$$

The empty state s_0 does not update the internal state of the functional entity. $FunM_{fe}(\phi,s)$ combines new internal state with a perception and previous internal state $FunM_{fe}: :\Phi_{fe}\times S_{fe}\rightarrow S_{fe}$.

Once the functional entity has determined the following action, it must execute. The actions on the system are made by the execution function defined as $Exec_{fe}:P\times\Phi_{fe}\rightarrow\Gamma$, so that $\gamma_{fe}=Exec_{fe}(p,\phi_{fe})$. Taking into account the definitions so far, the function $Exec_{fe}$ is defined as:

$$Exec_{fe}(p,\phi_{fe})=post \quad \text{if } PreE_{fe}(p,\phi)=True$$
$$Exec_{fe}(p,\phi_{fe})=\gamma_0 \quad \text{if } PreE_{fe}(p,\phi)=False \qquad (13)$$

Figure 1 shows a graphical representation of a *fe*, with the component elements and internal signals.

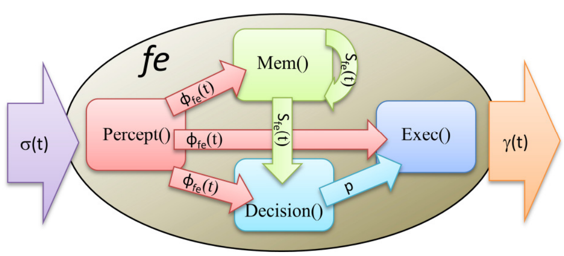


Figure 1. Graphical representation of a functional entity and its components

Finally, the mechanical system (MS) is defined by the set of all physical devices as sensors (S) and actuators (A) that form the robot $ASD=\langle a_1,a_2,\ldots,a_n,s_1,s_2,\ldots,s_m\rangle$ and the reaction function that describes how the robot reacts in front of influences, $MS=\langle ASD,React_{MS}\rangle$.

In the case of the robotic system is not necessary to model the behavior of the function *React*. We get the answer directly from the real robot or simulated robot. Thus, the answer depends on real physical laws and not of a model. Although we are modeling the control system, it must get answers from the physical system that controls and not an abstract model of robot. In this way, we avoid modeling worlds partially, allowing the system to interact directly with the reality around him and not abstract entities.

The system aims to transform the state of the world to a new state using the action and this transformation is seen as the system's reaction to attempts to influence from the functional elements, ie *React:* $\Sigma \times \Gamma \rightarrow \Sigma$. The new state of the system can be obtained from:

$$\sigma(t+1)=React(\sigma(t), \cup^{\Gamma}(\gamma_1,\gamma_2, \ldots,\gamma_n)) \qquad (14)$$

The empty influence is the neutral element of React. Since the implementation of the action gives the influence empty, it can also be seen as the neutral element, and will not cause system state change. The dynamics of the system would be defined by:

$$\begin{cases} \sigma(t+1)=React_{MS}(\sigma(t), {}^{n}\prod_{i=1} Exec_i(Decision_i(\phi_i(t),si(t)), \phi_i(t))) \\ s_1(t+1)=Mem_1(\phi_1(t),s1(t)) \\ \ldots. \\ s_n(t+1)=Mem_n(\phi_n(t),s_n(t)) \\ \qquad con\ \phi_i(t)=Percept_i(\sigma(t)) \end{cases} \qquad (15)$$

In Figure 2 we can see a graphical representation of the full model.

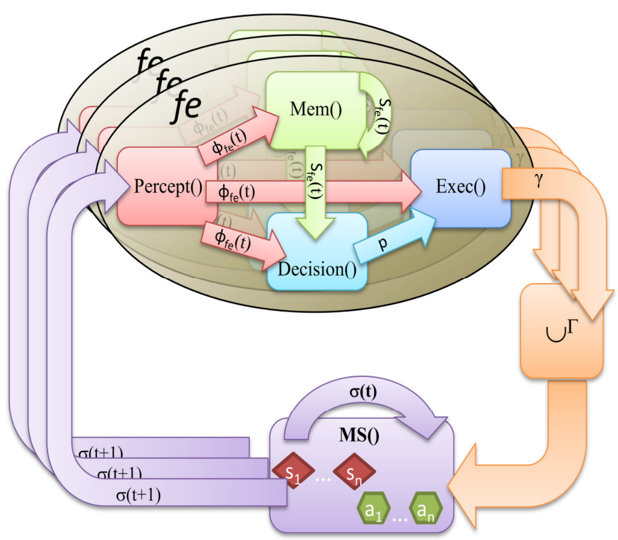


Figure 2. Control system consists of several *fe* and interacting with the mechanical system through the union of all its influences

Our model does not make a division between reactive and deliberative activity because the human nervous system does not. Each *fe* develops an activity that can be reactive, deliberative or hybrid. The agent paradigm gives us the ability to create this hybrid architecture. This allows the model to take advantage of the approaches and techniques in both fields. The model does not include communications, these will be resolved by the framework and remain hidden for robot logic. It also provides the distributed nature of biological system and its overall functioning, have specialized centers for each task and that the union of simple actions contribute a complex emergent behavior [8].

4. SOFTWARE TECHNOLOGIES FOR IMPLEMENTATION

The proposed model reveals a set of requirements that determine the software paradigm chosen for implementation. These are:

Autonomy

Each of the control center has autonomous character, each entity is perceiving the state of the world in which he is interested. This perception can be done about the outside world or the status of other centers and this behavior implies that the operation of system is asynchronous.

Flexibility

Our model needs a great deployment flexibility, ie, to modify, add or remove elements and their relationships that allows us to modify the resulting emergent behavior. As has happened in the nervous system and its evolution over the years, each time you add new functionality, improved behavior or the system requirements change, the interconnections are altered, new centers and some centers are disabled or have a more minor influences.

Decoupling

This decoupling has to happen in two ways. On the one hand allow functional decoupling between entities, ie software paradigm hides the details of the communication layer between the elements, allowing well-defined relationships freely and allow asynchronous nature of the system.

On the other hand, technology must also allow the independence of the underlying hardware such as computers resources or sensing/actuating devices. In this way a control system may be used in any robot, changing only the functional elements that form and their interconnections.

Reuse

The development platform should enable code reuse. Each functional entity will be implemented independently and then linked into an integrated system, and therefore the platform should be allowed to use the existing modules.

Interoperability

For the distributed nature of the system and the above features, the communication system must allow relations between modules that may be physically located on different nodes. In this way we could use functions developed by other teams, increasing the possibilities of exploiting the know-how of many researchers.

In our work, we use a service-oriented architecture (SOA) because the principles of this architecture align perfectly with the requirements demanded by our model [33][34]. Commonly highlighted features of the services are [33][34][35]: autonomous entities and self-contained coarse grain, abstract, reusable, durable, well-defined contracts, interoperable, modular, loosely coupled, discoverable, aligned with the business and interoperable. It is for these reasons that SOA provide us, on our conceptual model, the architectural principles that allow us to mitigate some of the biggest handicaps found in robotics such as the strong coupling between physical and logical levels and the rigidity of the systems to changes or evolution. Moreover, relying on a widespread and studied paradigm allows us to leverage the SOA community experience, and have solutions based on best practices and successful proposals.

5. TESTING

To validate our proposal, we implemented various behaviors on mobile autonomous robots, following the model proposed. The implementation process is to break the desired behavior in the fundamental tasks to be carried out and describe each of them using the model, characterizing each of the functions and relationships between them, determining the afferent and efferent signals of each functional entity. Mobile systems are especially interesting when they are treated in open and dynamic environments, because in such environments, the quantity, quality and accuracy of information is uncertain [36], because we cannot cover all possible situations and circumstances and therefore cannot develop full models of the world for the control system. Another reason to address this type of system is: robots can be very diverse, ie they may use different motor systems (legs, wheels, chains) different sensory systems, multiple algorithms for estimation of position, route calculation, ETC. which implies that vary the sources of information and therefore requires great flexibility and adaptability of the system. It is also possible to alter the desired behaviors such as scrolling through the environment, goal seeking, avoidance of obstacles and dangers, and so on., Which means more or less tasks involved.

In our work we have addressed two behaviors: Behavior 1 (B1) - navigating through the environment from a point of origin to a destination, and Behavior 2 (B2) - navigating through the environment with obstacle avoidance. As we shall see, given the characteristics of the model, B2 will be implemented by incorporating new elements on B1. For our system we have contemplated a generic robot equipped with two actuators (right wheel and left wheel) from which we get the current position of the wheel (shaft encoder sensor), a digital compass that indicates the current direction and a front-sensor obstacle detection (Figure 3). The type of robot will change the number of entities involved in the system, but only affects to entities that have contact with the physical devices.

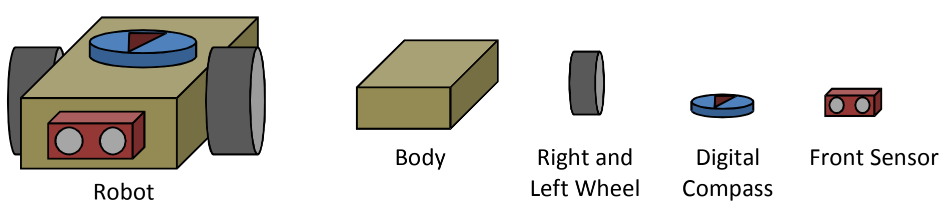


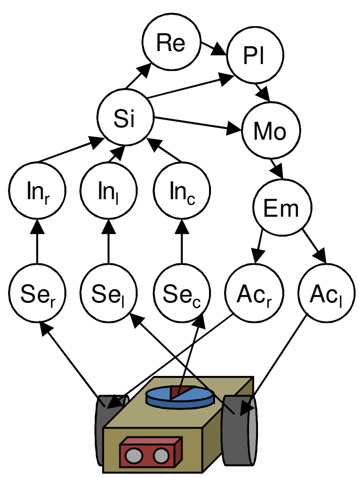
Figure 3. Scheme of the robot used for testing

In the functional analysis of behavior we have divided each task of the robot in a functional entity, separating and decoupling each task. Each entity will be carried out independently.

B1 analysis produces the following elements: *Sensorization*, responsible for monitoring the sensing devices, *Interpretation*, responsible for translating the values obtained in consistent data (eg floating numbers to two decimal numbers), *Situation*, responsible for using data interpretation to obtain an estimate of the robot's position (in this case position in the environment, but could be other estimates as the position of a robotic arm or relative position): *Reasoning*, element responsible for determining the mission to perform, in this case lead the robot from point A to point B, *Planner*, responsible for planning the route of the robot, *Movement*, transform the planned route to movements, *Embodiment*, responsible for transforming the type of motion in terms of physical structure the robot (for example, moving forward would become both wheels forward); *Actuator*, adapt and transmit orders to the actuating devices. Figure 4 shows the elements that make up B1 and their relationships.

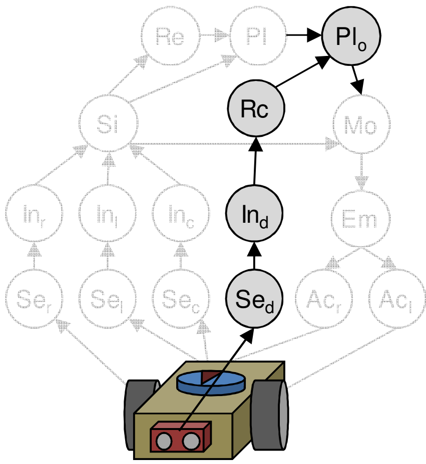
B2 analysis adds new centers to B1: *Sensorization* to control the distance sensor, *Interpretation* to transform distance value to meters, a new element called *Restriction* responsible for calculating where is a obstacle, a new element planner (called *PlannerObs*) that modifies outputs from Planner to avoid obstacles. Figure 5 shows this new elements added to B1.

Each of the control system entities perform specific functions, simple and decoupled from other functions. *Situation*: estimates the current position using odometry techniques, combining the distances travelled by each wheel. *Interpretation*: translate wheel encoder value to distance using the diameter of every wheel. *Planner*: decide the speed and direction to follow depending on the current position and target position. Separate each system function in a functional entity allows us to modify an element of the robot easily, for example the diameter of a wheel, this change only causes changes in the sensing elements while maintaining the independence of the rest of the system



Task	Description
Se_r	Get rigth wheel position
Se_l	Get left wheel position
Se_c	Get value of digital compass
In_r	Transform right wheel position to distance
In_l	Transform left wheel position to distance
In_c	Transform compass value to degree
Si	Estimate position
Re	Determine destination
Pl	Plan route to follow
Mo	Transform route to movement
Em	Transform movement to actions of actuators
Ac_r	Transform action to right motor signal
Ar_l	Transform action to left motor signal

Figure 4. Decomposition of B1, functional entities and their relationships



Task	Description
Se_d	Get distance value of distance sensor
In_d	Transform value of distance to meters
Rc	Calculate distante to obstacle
Pl_o	Modify the planned route to avoid obstacles

Figure 5. B2 new elements and new relations between elements of B1 and elements of B2

In addition, to construct new systems can reuse the elements thereby facilitating rapid development and low cost. The algorithm chosen to implement each function are simple, but you can use any other more complex algorithm.

The implementation of each entity has been created using the Microsoft Robotics Developer Studio (MRDS). MRDS provides an integrated .NET development environment for the design, execution and debugging scalable robot applications, concurrent and distributed in addition to providing features such as service coordination, monitoring, configuration, deployment and reuse [37]. This platform allows us to implement each functional entity as a service with low coupling

and a model like behavior. In addition it also has a powerful environment for composing application. Using this environment, we can indicate the services and their relationships easily. For testing we used the MRDS simulator and Lego robots because our intention is not build efficient robots, is to test the features of flexibility, composition, emergent behavior, and so on., described above. Figure 6 shows the robot implemented in the simulator and the Lego robot used for testing.

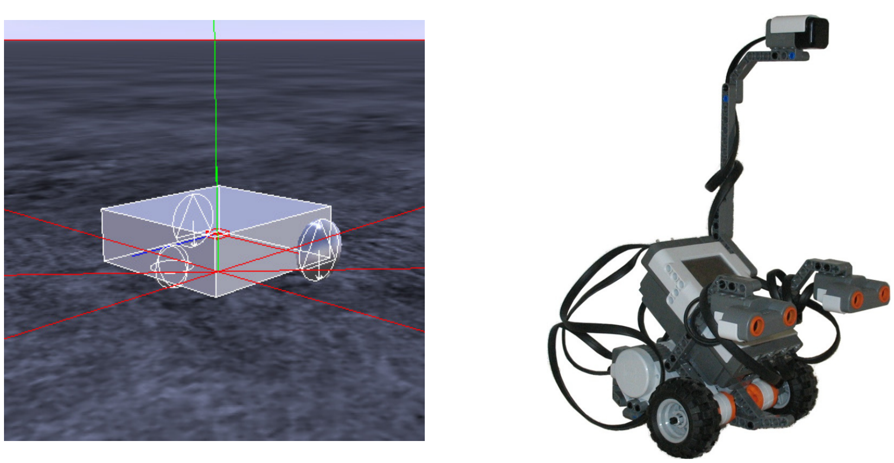


Figure 6. Simulated robot and Lego robot used for testing.

Following the implementation of services and the composition of the control system, we can see the robot, both real and simulated, is capable of carrying out the behaviors described. If robots use B1, robots move from the point of origin to destination point. If they use B2, robots also prevent obstacles in its path. To use B1 or B2 is only necessary to add more or less control system elements. To use a simulated robot or a real robot you just have to modify the connections of the sensing and actuator elements. This modification is easily done at MRDS, because the composition of services is simply add or remove services and relationships because services are low coupling.

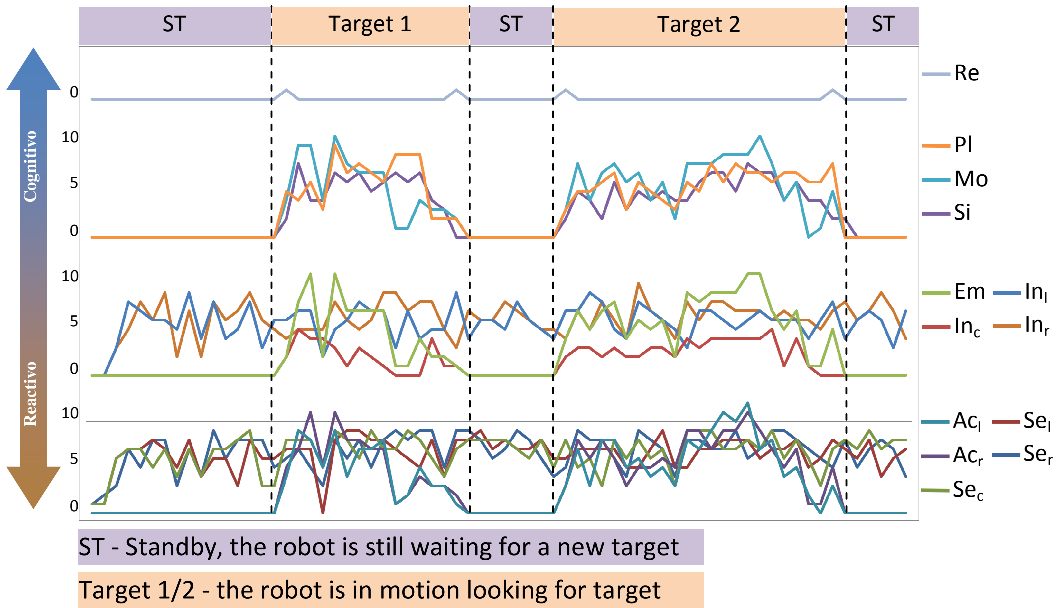


Figure 7. Graph showing the activity of each center during the behavior B1 looking for 2 goals. The activity is measured in number of messages sent per second

Each service is independent, uses its own frequency, its execution can influence or not the rest of the system and the operation is completely asynchronous, using the message passing between each service. In Figure 7 we can observe the activity of the control entities during the execution of B1.

The graph shows how the entities closest to the physical devices (sensorization, actuator and interpretation) are very active, even when the robot is not looking for the destination location. The entities which operate on a higher level, more cognitive, have lower activity. The most extreme case is the entity *Reasoning*, that is only active in 4 times to start and stop the robot. The performance of B1 and B2 with real or simulated robots get similar graphs.

Figure 8 shows images of the robot running B2. B2 uses the same entities that B1, plus the entities described for B2. The activity of the service follows the same pattern as B1.

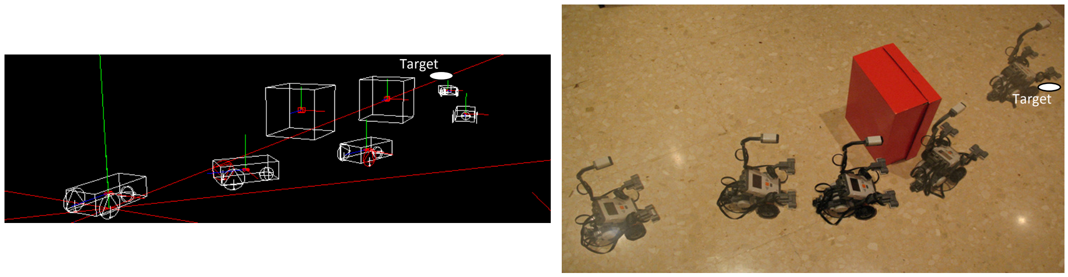


Figure 8. Simulated and Lego robot executing B2

Each entity is a separate service, and each service can be physically located on a different node network, thus maintaining the scalability of the system and make better use of resources. More complex services can be located in nodes with greater computing resources, while the lighter services can be placed on embedded devices.

If we compare the system with a reactive action selection, we see that here is not necessary to select the appropriate behavior, because all behaviors are running simultaneously. If we compare the system with a deliberative, we see that the robot interacts with the real world, the system does not require complex models of the world and rules. The distributed control system is like the nervous system, and can perform complex behaviors. The system thus provides a hybrid architecture. But the system does not create individual elements for each entity, all are homogeneous. The flexibility of the model allows an entity reactive, deliberative, or any combination.

6. Conclusions and Future Work

In this paper we have proposed a control system model inspired by the functional pillars of human neuroregulatory system, characterizing the control centers through a formal framework based on multi-agent systems. Having a formal model allows us to have a common reference point for the construction of any robotic control system. This model has the advantages of both reactive and deliberative architectures because it is a hybrid system like the human neuroregulatory.

Has been justified the need for a software paradigm that allows to implement the features of the model, without new technologies or frameworks and exploiting existing ones. Specifically, we selected the services because the nature of the model is distributed and services have demonstrated their maturity and ability in other environments such as e-business, in addition to adapt perfectly to the structure of the modelled entities.

A concrete implementation of the model for the control of mobile systems in open and dynamic environments has been provided. This implementation allows to demonstrate the feasibility of the proposal, the flexibility of the model and the suitability of selected technology. The examples used can incorporate more or fewer centers, change behavior and modify the devices, whether virtual or real.

During testing, the model exhibits the characteristics of the regulatory system: services closer to the physical devices maintain a constant activity (reactive), services that exhibit more cognitive functions become operational only when circumstances require it (deliberative). Overall the system shows a hybrid architecture but does not separate the types of activity in layers. This behavior, hybrid and asynchronous, is characteristic of biological systems and is wanted in bio-inspired robotics.

Currently we are working on two main lines. On the one hand we are increasing the range of services available to provide functionality, such as environment mapping, the positioning error correction by integration of several signals and pattern recognition in images. On the other hand, we are also investigating the incorporation of techniques of neuromodulation on the system. Neuromodulation not alter the functions of the entities but increase or decrease its intensity, modifying the emergent behavior.

REFERENCES

[1] Bekey, G., Ambrose, R., Kumar, V., Lavery, D., Sanderson, A., Wilcox, B., Yuh, J., Zheng, Y. (2008). "Robotics. State of the art and future challenges". *Imperial College Press*.

[2] International Federation of Robotics Statistical Department. (2010). "World Robotics: Service Robots 2010".

[3] Berná-Martínez, J. V., Maciá-Pérez, F. (2010). "Model of Integration and Management for Robotic Functional Components Inspired by the Human Neuroregulatory System". *IEEE International Conference on Emerging Technologies and Factory Automation 2010*. Bilbao-Spain.

[4] Brooks, R. A. (2002). "Flesh and machines. How robots will change us". *Pantheon*.

[5] Langley, P., Laird, J. E., Rogers, S. (2009). "Cognitive architectures: Research issues and challenges". *Cognitive Systems Research*, 10-2, pp.141-160. Elsevier.

[6] Redgrave, P., Prescott, T.J., Kurney, K. (1999). "The basal ganglia: a vertebrate solution to the selection problem?". *Neuroscience* 89:1009–1023.

[7] Arkin, R. C. (1998). "Behavior-Based Robotics". *The MIT Press*.

[8] Brooks, R. A, (1986). "A robust layered control system for a mobile robot". *IEEE Journal of Robotics and Automation*, 2(1), 14-23.

[9] Arkin, R.C. (2008). "Governing lethal behavior: embedding ethics in a hybrid deliberative/reactive robot architecture". *3rd ACM/IEEE ICHRI*.

[10] Bullo, F., Cortes, J., Martinez, S. (2009). "Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms". *Princeton University Press*.

[11] Anderson, J. R., Bothell, D., Byrne, N. D., Douglass, S., Lebiere, C., Qin, Y. (2004). "An integrated theory of the mind". *Psychological Review*, 111, 1036–1060.

[12] Laird, J. E. (2008). "Extending the Soar cognitive architecture". *Proc. of the artificial general intelligence conference*. Memphis, TN: IOS Press.

[13] Langley, P., Messina, E. (2004). "Experimental studies of integrated cognitive systems". *Proc. of the performance metrics for intelligent systems workshop.*

[14] Carbonell, J. G., Knoblock, C. A., Minton, S. (1990). "PRODIGY: An integrated architecture for planning and learning". In K. Van Lehn (Ed.), *Architectures for intelligence*. Hillsdale, NJ: Lawrence Erlbaum.

[15] Zhou, T., Fan, X., Yang, S. (2010). "Path planning for mobile robots based on hybrid architecture platform". *Computer and Information Science*, 3(3), pp 117-121.

[16] Posadas, J. L., Poza, J. L., Simo, J. E., Benet, G., Blanes, F. (2008). "Agent-based distributed architecture for mobile robot control". *Engineering Applications of Artificial Intelligence*, 21, pp. 805-823.

[17] Li, C., Bodkin, B., Lancaster, J. (2009). "Programming Khepera II Robot for autonomous navigation and exploration using the hybrid architecture". *Proceedings of the 47th Annual Southeast Regional Conference*.

[18] Noback, C. R., Strominger, N. L., Demarest, R. J., Ruggiero, D. A. (2005). "The Human Nervous System. Structure and Function". *Williams & Wilkins*.

[19] Furness, J. B (2006). "The Enteric Nervous System". *Blackwell Publishing*.

[20] Jackson, J. H. (1958). "Evolution and dissolution of the nervous system". In J. Taylor (Ed.), *Selected writings of John Hughlings Jackson*, 2, pp. 45–75. London: Staples Press. (Original work published 1884).

[21] Le Doux, J. (1995). "The Emotional Brain. Emotion: clues from de brain". *Annual Review of Psychology*, 46, pp. 209-235.

[22] Berntson, G. G., Boysen, S. T., Cacioppo, J. T. (1993). "Neurobehavioral organization and the cardinal principle of evaluative bivalence". *Annals of the New York Academy of Sciences*, 702, 75-102.

[23] Gallistel, C. R. (1982). "The organization of action: a new synthesis". *Psychology Press*.

[24] Arnold, A. P., Etgen, A. M., Fahrbach, S. E., Rubin, R. T., Pfaff, D. W. (2009). "Hormones, Brain and Behavior". *Academic Press*.

[25] Ferber, J. (1999). "Multi-Agent Systems. An Introduction to Distributed Artificial Intelligence". *Addison-Wesley*.

[26] Weiss, G. (1999). "Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence". *The MIT Press*.

[27] Macia-Perez, F., Garcia-Chamizo, J. M. (2006). "Mobile Agent System Framework Suitable for Scalable Networks". *Kybernetes. The International Journal of Systems and Cybernetics*, 35, 688-688. Emerald.

[28] Engström, H., Kummeneje, J. (1997). "ABBility: Agent Technology and Process Control". *Dep. of Computer and Systems Science*. University of Stockholm.

[29] Mas, A. (2005). "Agentes Software Y Sistemas Multi-Agente: Conceptos, Arquitecturas y Aplicaciones". *Pearson*.

[30n] Shetty, S., Kiran, S., Babu, M., Singh, S. " Symbolic Verification of a Strength Based Multi-Agent Argumentation Scheme". International Journal of Artificial Intelligence & Applications (IJAIA), Vol.2, No.2, April 2011.

[31] Garcia-Chamizo, J. M., Macia-Perez, F., Soriano-Paya, A., Ruiz-Fernandez, D. (2003). "Simulation of the Neuronal Regulador of the Coger Urinary Tract Using a Multiagent System". *LNCS, Springer-Verlag*, 2687, 591-598.

[32] Ruiz-Fernandez, D., Garcia-Chamizo, J. M., Macia-Perez, F., Soriano-Paya, A. (2004). "Modeling the Distributed Control of the Lower Urinary Tract Using a Multiagent System". *Lecture Notes in Artificial Intelligence*, 3131, 104 -114.

[33] Erl, T. (2005). "Service-Oriented Architecture (SOA): Concepts, Technology, and Design". *Prentice Hall*.

[34] Marks, E. A., Bell, M. (2006). "Service-Oriented Architecture. A planning and Implementation Guide for Business and Technology". *Wiley*.

[35] SOA Reference Model. OASIS. (Online) Available at: http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf. Sep, 2011.

[36] Della-Penna, G., Intrigila, B., Magazzeni, D., Mercorio, F. " Resource-Optimal Planning For An Autonomous Planetary Vehicle". International Journal of Artificial Intelligence & Applications (IJAIA), Vol.1, No.3, July 2010.

[37] Johns, K., Taylor, T. (2008). "Professional Microsoft Robotics Developer Studio". *Wrox*.

Authors

Jose Vicente Berna-Martinez was born in Spain in 1978. He received his engineering degree and the Ph.D. degree in Computer Science from the University of Alicante in 2004 and 2011 respectively. Since 2006, he is an Associate Professor at the University of Alicante. His research interests are in the area of computer networks, distributed systems, bio-inspired systems and robotics which are applied to industrial problems.

Francisco Macia-Perez was born in Spain in 1968. He received his engineering degree and the Ph.D. degree in Computer Science from the University of Alicante in 1994 and 2001 respectively. He worked as System's Administrator at the University of Alicante form 1996 to 2001. He was an Associate Professor from 1997 to 2001. Since 2001, he is an Assistant Professor and currently he is the Director of the Department of Computer Science and Technology at the University of Alicante. His research interests are in the area of network management, computer networks, smart sensor networks and distributed systems, which are applied to industrial problems.